

Déployez votre application sur un serveur comme un pro !

[Application déployée sur AWS](#)

[Tableau Trello](#)

[Repo Github](#)


Décomposition du `settings.py` pour les différents environnements

Création de settings pour:

- [le développement \(version de base\)](#)
- [Heroku](#)
- [AWS](#)
- [Pytest](#)
- [Travis](#)

Déploiement

Création d'une instance EC2 sur AWS

**Ubuntu Server 18.04 LTS (HVM), SSD Volume Type** - ami-0c55b159cbfafa1f0 (64 bits x86) / ami-0f2057f28f0a44d06 (64 bits Arm)
Sélectionner

Eligible à l'offre

64 bits (x86)
64 bits (Arm)

Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Type de périphérique racine: ebs Type de virtualisation: hvm ENA activée: Oui

Modification du "Groupe de sécurité" pour permettre l'entrée en HTTP (port 80)

Récupération du fichier *pem*

Connexion ssh:

```
ssh -i "~/AWS/zepman_air13.pem" ubuntu@ec2-3-16-90-115.us-east-2.compute.amazonaws.com
```

Mise à jour des dépôts et installation de pyenv (installation et activation de Python 3.7.2), postgresql, nginx et supervisor

```
sudo apt-get update && sudo apt-get upgrade
sudo apt-get install python3-pip python3-dev libpq-dev postgresql
postgresql-contrib nginx supervisor virtualenv
```

Copie du repo github et creation de l'environnement virtuel

```
git clone https://github.com/Zepmanbc/oc_dapython_pr10.git
cd oc_dapython_pr10
virtualenv env -p python3
source env/bin/activate
```

Création des variables d'environnement pouvoir faire la migration

```
export ENV=AWS
export DB_NAME=purbeurre
export DB_USER=pb_sql_user
export DB_PASSWORD=*****
export SECRET_KEY=[SECRET_KEY]
export DJANGO_SETTINGS_MODULE=purbeurre.settings.aws_settings
```

Installation des dépendences

```
pip install -r config/requirements.txt
```

le fichier a été obtenu avec cette commande

```
pipenv run pip freeze > config/requirements.txt
```

Configuration de postgres

Creation d'un utilisateur et d'une base dans postgres

```
sudo -u postgres psql
CREATE DATABASE purbeurre;
CREATE USER pb_sql_user WITH PASSWORD '*****';
```

Modifications pour améliorer la performance des requêtes (cf tuto)

```
ALTER ROLE pb_sql_user SET client_encoding TO 'utf8';
ALTER ROLE pb_sql_user SET default_transaction_isolation TO 'read
committed';
ALTER ROLE pb_sql_user SET timezone TO 'Europe/Paris';
```

Donne les droits à l'utilisateur sur la base *purbeurre*

```
GRANT ALL PRIVILEGES ON DATABASE purbeurre TO pb_sql_user;
```

Collecte des fichiers *static*, migration, collecte de produits

```
python ~/oc_dap/python_pr10/purbeurre/manage.py collectstatic
python ~/oc_dap/python_pr10/purbeurre/manage.py migrate
python ~/oc_dap/python_pr10/purbeurre/manage.py fillindb 50
```

Création d'un super utilisateur

```
python ~/oc_dap/python_pr10/purbeurre/manage.py createsuperuser
```

Configuration de *Nginx*

Fichier de configuration Nginx (config/nginx/purbeurre.conf)

```
server {

    listen 80;
    server_name 3.16.90.115;
    root /home/ubuntu/oc_dap/python_pr10/purbeurre/

    location / {
        proxy_set_header Host $http_host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_redirect off;
        if (!-f $request_filename) {
            proxy_pass http://127.0.0.1:8000;
            break;
        }
    }

    location /static {
        alias /home/ubuntu/oc_dap/python_pr10/purbeurre/staticfiles/;
    }
}
```

Création du lien symbolique vers le fichier de configuration puis rechargement de nginx

```
sudo ln -s ~/oc_dapython_pr10/config/nginx/purbeurre.conf /etc/nginx/sites-enabled
sudo service nginx reload
```

Configuration de *Supervisor*

Création du fichier de [configuration de supervisor](#)

```
[program:purbeurre]
directory=/home/ubuntu/oc_dapython_pr10/
command=/home/ubuntu/oc_dapython_pr10/env/bin/gunicorn --chdir purbeurre
purbeurre.wsgi:application
autostart = true
autorestart = true
user=ubuntu
environment = ENV=AWS,
               DB_NAME=purbeurre,
               DB_USER=pb_sql_user,
               DB_PASSWORD=*****,
               SECRET_KEY=[SECRET_KEY],
               DJANGO_SETTINGS_MODULE=purbeurre.settings.aws_settings

[supervisord]
environment=LC_ALL='en_US.UTF-8', LANG='en_US.UTF-8'
```

Création d'un lien symbolique vers le dossier de configuration et prise en compte de la nouvelle configuration

```
sudo ln -s ~/oc_dapython_pr10/config/supervisor/purbeurre-gunicorn.conf
/etc/supervisor/conf.d/
sudo supervisorctl reread
sudo supervisorctl update
sudo supervisorctl status
```

Mise en place de *Travis*

Fichier de configuration

[.travis.yml](#)

Interface de Travis

The screenshot displays the Travis CI dashboard for the repository **Zepmanbc / oc_dapython_pr10**. The top navigation bar includes links for Dashboard, Changelog, Documentation, and Help. On the left, a sidebar lists recent builds for the repository, including **oc_dapython_pr10** (build #11), **oc_dapython_pr8** (build #26), **oc_dapython_pr3** (build #26), and **oc_dapython_pr7** (build #42). The main area shows the details of the current build for the **master** branch. It indicates that the build is **passing** and shows the commit **bf1cfd1** by **Zepmanbc**. The job log at the bottom provides a detailed view of the build process, including the installation of Firefox and the cloning of the repository.

Et ajout de [coveralls](#)

Monitoring

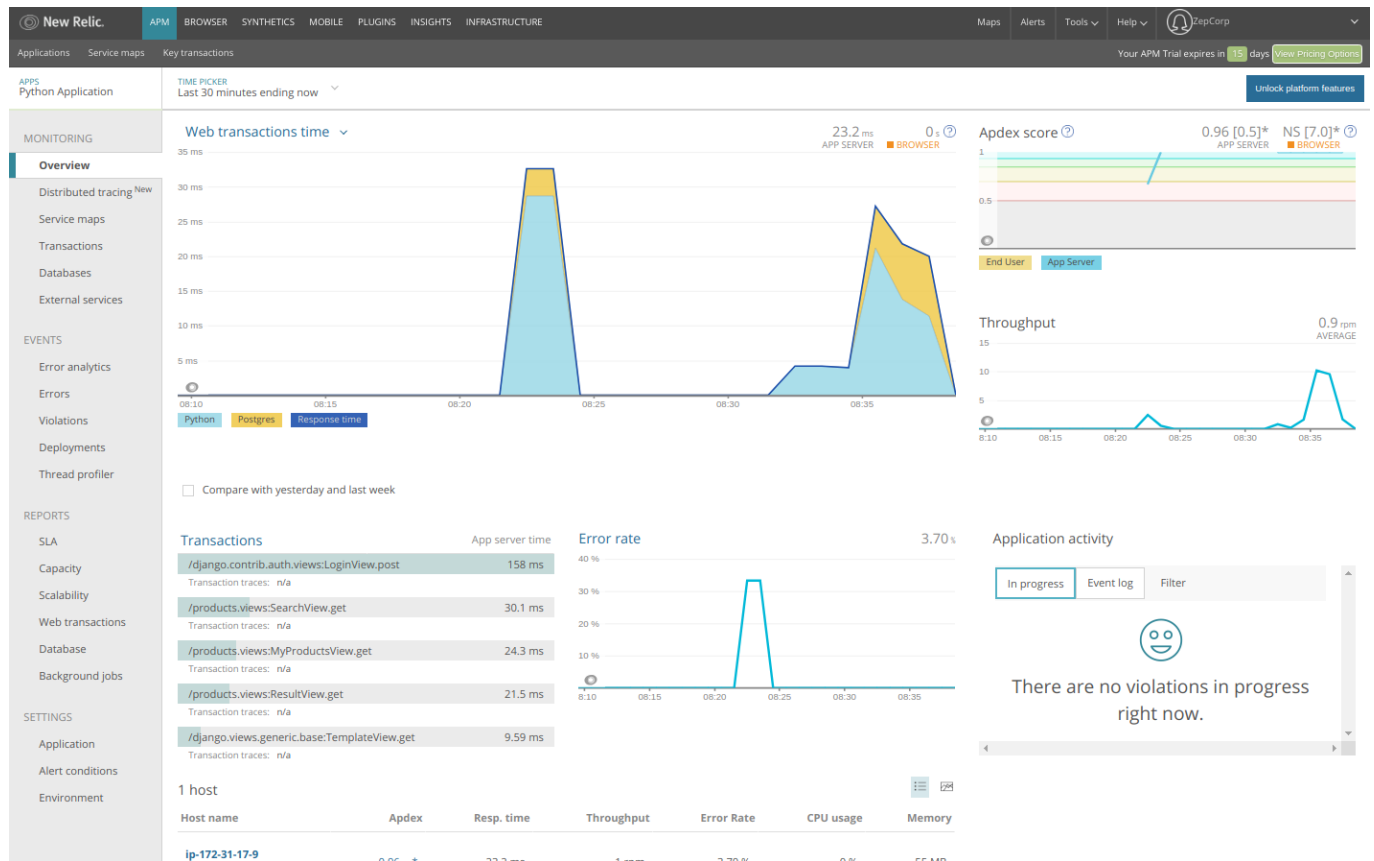
Mise en place de *Newrelic*

Création du fichier de configuration

```
newrelic-admin generate-config <your-key-goes-here> newrelic.ini
```

Modification de la commande de démarrage dans supervisor

```
command = /home/ubuntu/oc_dapython_pr10/env/bin/newrelic-admin run-program
/home/ubuntu/oc_dapython_pr10/env/bin/gunicorn --chdir purbeurre
purbeurre.wsgi:application
environment =
NEW_RELIC_CONFIG_FILE=/home/ubuntu/oc_dapython_pr10/newrelic.ini
```



Mise en place de Sentry

installation du package

```
pip install --upgrade sentry-sdk==0.7.11
```

Ajout de la configuration dans `settings/aws_settings.py`

```
import sentry_sdk
from sentry_sdk.integrations.django import DjangoIntegration

sentry_sdk.init(
    dsn="https://4cc8d88b64444388a3fbbc92cf017305@sentry.io/1445027",
    integrations=[DjangoIntegration()]
)
```

Exemple avec une url modifiée

<https://sentry.io/share/issue/6e936250d22a4019a08309cadfbc2f17/>

Mise en place du login sur les recherches des utilisateurs

[purbeurre/products/views.py](#)

```
from sentry_sdk import capture_message, configure_scope
(...)
class SearchView(ListView):
(...)
def get_queryset(self):
    with configure_scope() as scope:
        if '_auth_user_id' in self.request.session._session:
            scope.user = {
                "email":
User.objects.get(pk=self.request.session._session['_auth_user_id']).email
            }
            scope.level = 'info'
            capture_message('New search')

    return Product.objects.filter(
        product_name__icontains=self.request.GET['query']).\
        order_by('product_name')
```

Automatisations de la mise à jour des produits

Script qui lance la mise à jour des données : *update_job.sh*

```
source ~/oc_dapython_pr10/env/bin/activate
echo $(date) >> ~/log.txt
python ~/oc_dapython_pr10/purbeurre/manage.py fillindb 0 >> ~/log.txt 2>&1
```

Rendre exécutable le script

```
chmod +x config/update_job.sh
```

Tache *cron* qui s'exécute toutes les semaines

```
@weekly ~/oc_dapython_pr10/config/update_job.sh
```

le crontab sur le serveur et le log avec 1 produit qui a été mis à jour

```
ubuntu@ip-172-31-17-9:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
@weekly ~/oc_dapypthon_pr10/config/update_job.sh
ubuntu@ip-172-31-17-9:~$ cat log.txt
Mon Apr 29 21:29:05 UTC 2019
Loading .env environment variables...
Update - Le Ravioli (Pur Bœuf, Farce au Bœuf)
Done
```